REMARKS

In an Office Action dated June 16, 2006, the Examiner rejected claims 1, 2 and 5-8 under 35 U.S.C. §102(b) as anticipated by Baumgartner (US 6,108,764); and rejected claims 3-4 and 19-22 under 35 U.S.C. §103(a) as unpatentable over *Baumgartner* in view of Carpenter et al. (US 6,115,804).

Applicants have amended all independent claims to clarify the claimed invention. In particular, the claims have been amended to clarify that there is a fixed, one-to-one correspondence between the slots in the device cache and the entries in the cache line state directory portion corresponding to the device cache. As amended, the claims are patentable over the cited art.

A brief background discussion is in order to appreciate applicants' invention. In a NUMA architecture, a respective discrete subset of main memory is usually associated with each node, processor or group of processors. Each processor or group of processors may additionally cache arbitrary portions of main memory. Where multiple caches exist in different nodes, cache coherency becomes a significant architectural issue. To support cache coherency, it is known to maintain some form of cache line state directory in each node. The cache line state directory maintains cache line state information for multiple cache lines, which are generally lines originating in the same node's subset of main memory. By accessing this directory, it is possible to determine whether copies of data in the subset of main memory may exist elsewhere in cache, and to infer certain state information.

The data in the cache line state directory is not necessarily current, but is sufficient for various purposes. In particular, when a line of data is loaded into a cache, a corresponding reference is placed in the cache line state directory, so that it will always be possible to find a

reference to a line which is in the cache. But the reverse is not necessarily true immediately if a line is overwritten or invalidated in a cache. In the latter case, the reference may linger in the cache line state directory, although due to any of various mechanisms it is eventually overwritten with new data. Usually, the cache line state directory is set associative, accommodating multiple entries for each hashed or otherwise encoded portion of an address used to access cache. The set associativity must be sufficiently large so that, when a line is loaded to cache, there will be an available entry in the associativity set corresponding to the cache line address for a reference to the loaded line. It is also possible to design variable sized associativity set directory structures.

Applicants observed that certain device caches, such as a cache for an I/O bridge device, tend to have a very high turnover in a relatively small cache which acts as a buffer, and as a result of the conventional set associative design, can rapidly displace a large number of references in a cache line state directory. This causes various inefficiencies in the use of cache line state directory space, in the need to send invalidation messages to the I/O bridge device, and so forth.

Since the cache in an I/O bridge device or certain similarly behaving devices is often quite small, applicants propose to address this problem by providing a separate cache line state directory portion, in which the entries do not correspond to real addresses (as in the case of a set associative cache), but instead have a one-to-one correspondence to the slots in the device cache. Thus, the number of references required to be kept for the device cache can never exceed the number of slots in the device cache (unlike the case of the conventional set associative cache).

Although applicants are unclear as to the Examiner's read of their claims on the apparatus disclosed in *Baumgartner*, upon review of the original claims, it appears that the aspect of a fixed one-to-one correspondence between the slots in the device cache and the entries in the cache line state directory was possibly unclear, and applicants have accordingly amended the claims to recite

that each contain a fixed number of slots or entries, and each entry has a fixed correspondence to a unique respective one of the slots.

All independent claims of applicants herein recite that a portion of the cache line state directory has a fixed one-to-one mapping of entries with slots in the device cache. Certain claims (e.g., independent claim 14 and dependent claims 3, 12 and 21) recite additionally that a separate portion of the cache line state directory contains entries for the processor cache or caches, and some of these claims recite additionally that entries in the processor portion correspond to real addresses of data (i.e., different indexing is used in different portions of the cache line state directory). None of these features are taught or suggested by the cited art.

Applicants' representative claim 1, as amended, recites:

1. A digital data processing system, comprising:
a memory;
at least one processor having at least one associated cache for temporarily caching data from said memory;
at least one device having a device cache, said device cache having a fixed number of slots for caching data, said fixed number being greater than one, each slot caching a cache line of data; and
a cache coherency mechanism, said cache coherency mechanism including a cache line state directory structure, said cache coherency mechanism selectively determining whether to send cache line invalidation messages to said at least one device using state information in said cache line state directory structure, wherein at least a portion of said cache line state directory structure associated with said at least one device contains exactly said fixed number of cache line entries, *each entry having a fixed correspondence to a unique respective one of said fixed number of slots* for caching data of said device cache. [emphasis added]

Conventionally, a cache line state directory is organized as a set associative directory, in which entries correspond to addresses. Both *Baumgartner* and *Carpenter* disclose what appears to be a cache line state directory or equivalent (which they call a "coherence directory"), but

neither discloses in detail the structure of the disclosed "coherence directory". In particular, neither discloses a coherence directory, in which directory entries correspond to ***slots*** in a cache. It is probable that their "coherence directory" is organized as a conventional set associative directory. But in any case, there is no need to speculate about what *Baumgartner* and *Carpenter* do not disclose, and do not even hint at. There is simply nothing in either reference that would suggest organizing the coherence directory so that entries correspond to slots in the cache.

Organizing a portion of the cache line state directory as entries having a fixed, one-to-one correspondence with cache slots effectively means that the number of such entries is the same as the number of cache slots, but it is not a mere number that is a significant feature. It is the fact that entries correspond to slots, and not to real addresses (as in the case of the conventional set-associative cache). Because entries correspond to slots, the number of entries containing device cache data can be limited to the relatively small number of slots of the device cache, and is inherently prevented from taking over large portions of a set associative cache, without requiring some auxiliary regulating mechanism.

For all the reasons stated above, the claims as amended are neither anticipated by, nor obvious over, *Baumgartner*, either alone or in combination with *Carpenter*.

In view of the foregoing, applicants submit that the claims are now in condition for allowance, and respectfully request reconsideration and allowance of all claims. In addition, the

Examiner is encouraged to contact applicants' attorney by telephone if there are outstanding issues left to be resolved to place this case in condition for allowance.

Respectfully submitted,

DUANE A AVERILL, et al.

By:_____
Roy W. Truelson
Registration No. 34,265

Telephone: (507) 202-8725